

LabVIEW™ Core 2 Course Manual

Course Software Version 2011
August 2011 Edition
Part Number 325292C-01

Copyright

© 1993–2011 National Instruments Corporation. All rights reserved.

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

For components used in USI (Xerces C++, ICU, HDF5, b64, Stingray, and STLport), the following copyright stipulations apply. For a listing of the conditions and disclaimers, refer to either the `USICopyrights.chm` or the *Copyrights* topic in your software.

Xerces C++. This product includes software that was developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright 1999 The Apache Software Foundation. All rights reserved.

ICU. Copyright 1995–2009 International Business Machines Corporation and others. All rights reserved.

HDF5. NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Copyright 1998, 1999, 2000, 2001, 2003 by the Board of Trustees of the University of Illinois. All rights reserved.

b64. Copyright © 2004–2006, Matthew Wilson and Synesis Software. All Rights Reserved.

Stingray. This software includes Stingray software developed by the Rogue Wave Software division of Quovadx, Inc. Copyright 1995–2006, Quovadx, Inc. All Rights Reserved.

STLport. Copyright 1999–2003 Boris Fomitchev

Trademarks

CVI, LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at ni.com/trademarks for other National Instruments trademarks.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents.

Sample

Worldwide Technical Support and Product Information

ni.com

Worldwide Offices

Visit ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

For further support information, refer to the *Additional Information and Resources* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the Info Code *feedback*.

Contents

Student Guide

A. NI Certification	v
B. Course Description	vi
C. What You Need to Get Started	vii
D. Installing the Course Software.....	vii
E. Course Goals.....	viii
F. Course Conventions	ix

Lesson 1

Common Design Techniques

A. Design Patterns	1-2
B. Single Loop Design Patterns.....	1-3
C. Multiple Loop Design Patterns.....	1-7
D. Timing a Design Pattern	1-11

Lesson 2

Synchronization Techniques

A. Notifiers	2-2
B. Queues	2-3

Lesson 3

Event Programming

A. Events.....	3-2
B. Event-Driven Programming.....	3-3
C. Caveats and Recommendations	3-11
D. Event-Based Design Patterns.....	3-11

Lesson 4

Error Handling

A. Importance of Error Handling.....	4-2
B. Detect and Report Errors	4-2
C. Errors and Warnings	4-3
D. Ranges of Error Codes	4-4
E. Error Handlers.....	4-5

Lesson 5

Controlling the User Interface

A. Property Nodes	5-2
B. Invoke Nodes	5-4
C. VI Server Architecture.....	5-5
D. Control References	5-6

Lesson 6

File I/O Techniques

A. File Formats	6-2
B. Binary Files	6-4
C. TDMS Files.....	6-12

Lesson 7

Improving an Existing VI

A. Refactoring Inherited Code.....	7-2
B. Typical Refactoring Issues.....	7-4
C. Comparing VIs.....	7-7

Lesson 8

Creating and Distributing Applications

A. Preparing the Files	8-2
B. Build Specifications	8-3
C. Building the Application and Installer.....	8-5

Appendix A

Additional Information and Resources

Glossary

Controlling the User Interface

When writing programs, often you must change the attributes of front panel objects programmatically. For example, you may want to make an object invisible until a certain point in the execution of the program. In LabVIEW, you can use VI Server to access the properties and methods of front panel objects. This lesson explains the Property Nodes, Invoke Nodes, VI Server, and control references.

Topics

- A. Property Nodes
- B. Invoke Nodes
- C. VI Server Architecture
- D. Control References

A. Property Nodes

Property Nodes access the properties of an object. In some applications, you might want to programmatically modify the appearance of front panel objects in response to certain inputs. For example, if a user enters an invalid password, you might want a red LED to start blinking. Another example is changing the color of a trace on a chart. When data points are above a certain value, you might want to show a red trace instead of a green one. Property Nodes allow you to make these modifications programmatically. You also can use Property Nodes to resize front panel objects, hide parts of the front panel, add cursors to graphs, and so on.

Property Nodes in LabVIEW are very powerful and have many uses. Refer to the *LabVIEW Help* for more information about Property Nodes.

Creating Property Nodes

When you create a property from a front panel object by right-clicking the object, selecting **Create»Property Node**, and selecting a property from the shortcut menu, LabVIEW creates a Property Node on the block diagram that is implicitly linked to the front panel object. If the object has a label, the Property Node has the same label. You can change the label after you create the node. You can create multiple Property Nodes for the same front panel object.

Using Property Nodes

When you create a Property Node, it initially has one terminal representing a property you can modify for the corresponding front panel object. Using this terminal on the Property Node, you can either set (write) the property or get (read) the current state of that property.

For example, if you create a Property Node for a digital numeric control using the Visible property, a small arrow appears on the right side of the Property Node terminal, indicating that you are reading that property value. You can change the action to write by right-clicking the terminal and selecting **Change To Write** from the shortcut menu. Wiring a False Boolean value to the Visible property terminal causes the numeric control to vanish from the front panel when the Property Node receives the data. Wiring a True Boolean value causes the control to reappear.

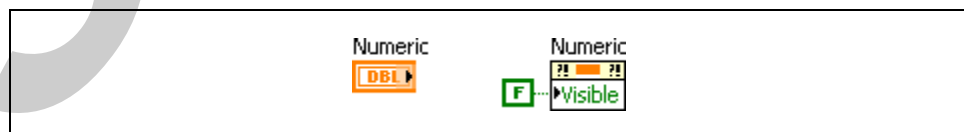


Figure 5-1. Using Property Nodes

To get property information, right-click the node and select **Change All to Read** from the shortcut menu. To set property information, right-click the node and select **Change All to Write** from the shortcut menu. If a property is read only, **Change to Write** is dimmed in the shortcut menu. If the small direction arrow on the Property Node is on the right, you are getting the property value. If the small direction arrow on a Property Node is on the left, you are setting the property value. If the Property Node in Figure 5-1 is set to Read, when it executes it outputs a True value if the control is visible or a False value if it is invisible.



Tip Some properties are read-only, such as the Label property, or write only, such as the Value (Signaling) property.

To add terminals to the node, right-click the white area of the node and select **Add Element** from the shortcut menu or use the Positioning tool to resize the node. Then, you can associate each Property Node terminal with a different property from its shortcut menu.



Tip Property Nodes execute each terminal in order from top to bottom.

Some properties use clusters. These clusters contain several properties that you can access using the cluster functions. Writing to these properties as a group requires the Bundle function and reading from these properties requires the Unbundle function. To access bundled properties, select **All Elements** from the shortcut menu. For example, you can access all the elements in the Position property by selecting **Properties»Position» All Elements** from the shortcut menu.

However, you also can access the elements of the cluster as individual properties, as shown in Figure 5-2.

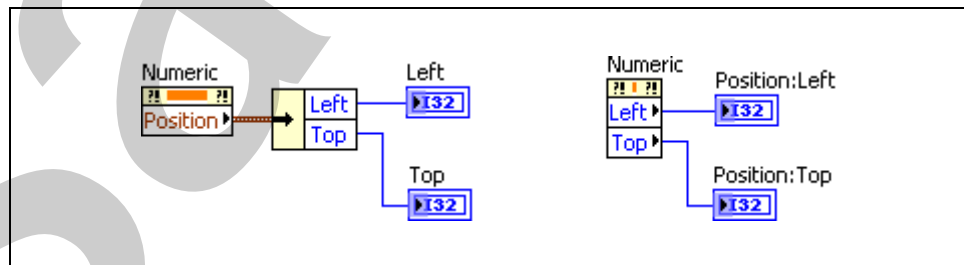


Figure 5-2. Properties Using Clusters

To practice the concepts in this section, complete Exercise 5-1.

B. Invoke Nodes

Invoke Nodes access the methods of an object.

Use the Invoke Node to perform actions, or methods, on an application or VI. Unlike the Property Node, a single Invoke Node executes only a single method on an application or VI. Select a method by using the Operating tool to click the method terminal or by right-clicking the white area of the node and selecting **Methods** from the shortcut menu. You also can create an implicitly linked Invoke Node by right-clicking a front panel object, selecting **Create»Invoke Node**, and selecting a method from the shortcut menu.

The name of the method is always the first terminal in the list of parameters in the Invoke Node. If the method returns a value, the method terminal displays the return value. Otherwise, the method terminal has no value.

The Invoke Node lists the parameters from top to bottom with the name of the method at the top and the optional parameters, which are dimmed, at the bottom.

Example Methods

An example of a method common to all controls is the Reinitialize to Default method. Use this method to reinitialize a control to its default value at some point in your VI. The VI class has a similar method called Reinitialize All to Default.

Figure 5-3 is an example of a method associated with the Waveform Graph class. This method exports the waveform graph image to the clipboard or to a file.

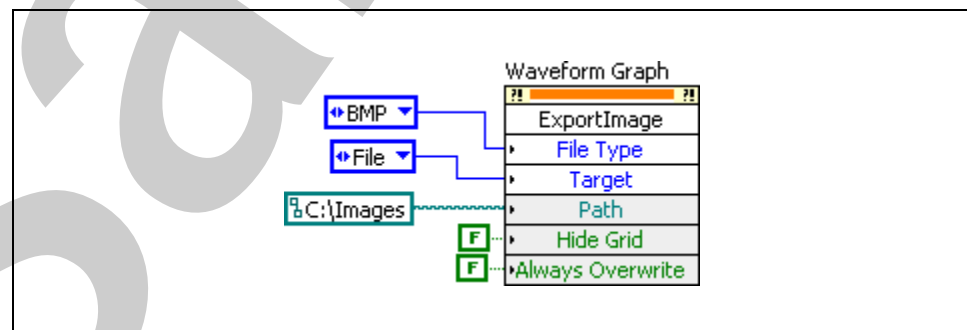


Figure 5-3. Invoke Node for the Export Image Method

To practice the concepts in this section, complete Exercise 5-2.

C. VI Server Architecture

The VI Server is an object-oriented, platform-independent technology that provides programmatic access to LabVIEW and LabVIEW applications. VI Server performs many functions; however, this lesson concentrates on using the VI Server to control front panel objects and edit the properties of a VI and LabVIEW. To understand how to use VI Server, it is useful to understand the terminology associated with it.

Object-Oriented Terminology

Object-oriented programming is based on objects. An *object* is a member of a class. A *class* defines what an object is able to do, what operations it can perform (methods), and what properties it has, such as color, size, and so on.

Objects can have methods and properties. *Methods* perform an operation, such as reinitializing the object to its default value. *Properties* are the attributes of an object. The properties of an object could be its size, color, visibility, and so on.

Control Classes

LabVIEW front panel objects inherit properties and methods from a class. When you create a Stop control, it is an object of the Boolean class and has properties and methods associated with that class, as shown in Figure 5-4.

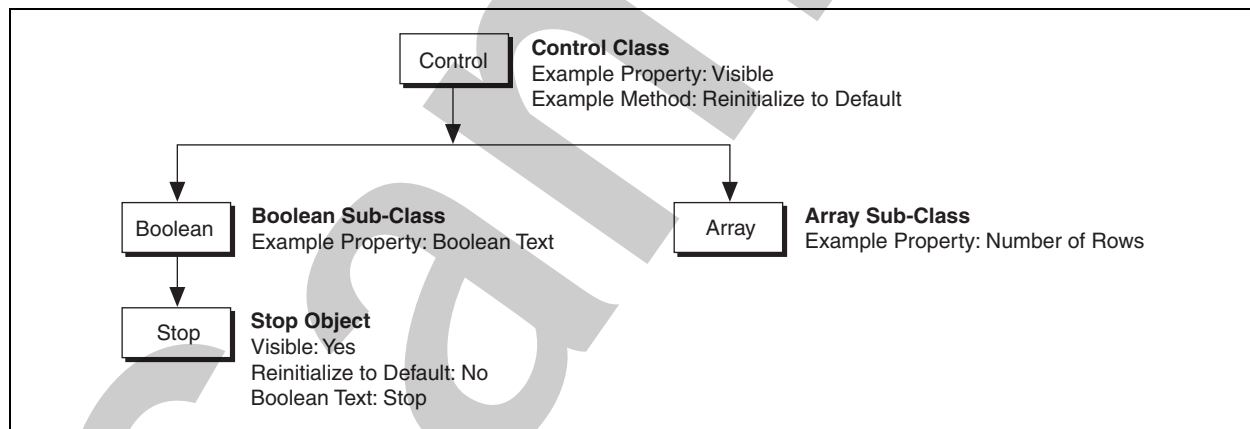


Figure 5-4. Boolean Class Example

VI Class

Controls are not the only objects in LabVIEW to belong to a class. A VI belongs to the VI Class and has its own properties and methods associated with it. For instance, you can use VI class methods to abort a VI, to adjust the position of the front panel window, and to get an image of the block diagram. You can use VI class properties to change the title of a front panel window, to retrieve the size of the block diagram, and to hide the **Abort** button.

D. Control References

A Property Node created from the front panel object or block diagram terminal is an implicitly linked Property Node. This means that the Property Node is linked to the front panel object. What if you must place your Property Nodes in a subVI? Then the objects are no longer located on the front panel of the VI that contains the Property Nodes. In this case, you need an explicitly linked Property Node. You create an explicitly linked Property Node by wiring a reference to a generic Property Node.

If you are building a VI that contains several Property Nodes or if you are accessing the same property for several different controls and indicators, you can place the Property Node in a subVI and use control references to access that node. A control reference is a reference to a specific front panel object.

This section shows one way to use control references. Refer to the *Controlling Front Panel Objects* topic of the *LabVIEW Help* for more information about control references.

Creating a SubVI with Property Nodes

As shown in Figure 5-5, the simplest way to create explicitly linked Property Nodes is to complete the following steps:

1. Create your VI.
2. Select the portion of the block diagram that is in the subVI, as shown in the first part of Figure 5-5.
3. Select **Edit»Create SubVI**. LabVIEW automatically creates the control references needed for the subVI.
4. Customize and save the subVI. As you can see in the second part of Figure 5-5, the subVI uses the default icon.

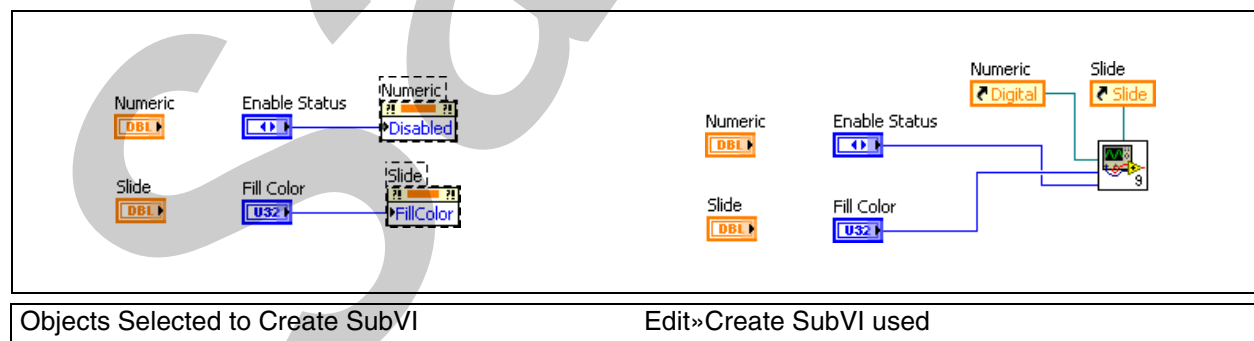


Figure 5-5. Using Edit»Create SubVI to Create Control References

Figure 5-6 shows the subVI created. Notice that the front panel Control Refnum controls have been created and connected to a Property Node on the block diagram.

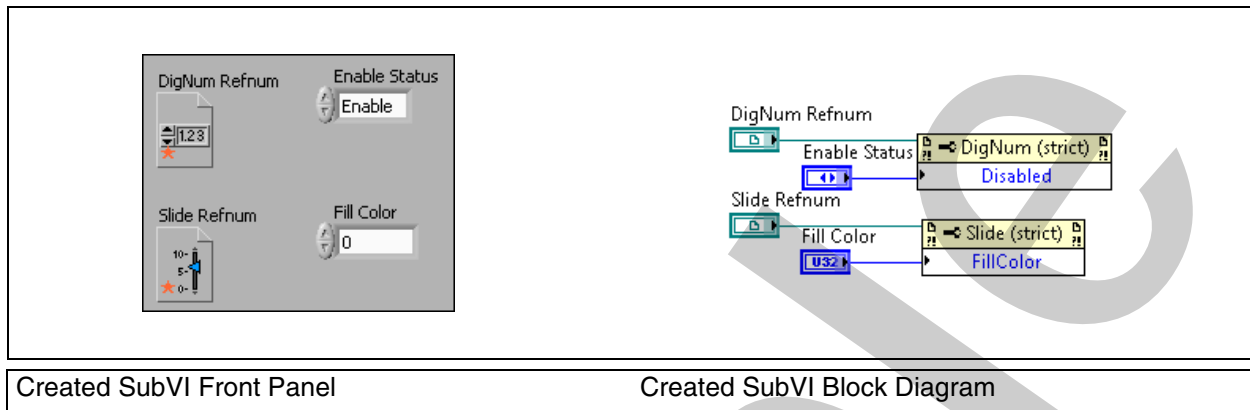


Figure 5-6. Sub VI Created Using Edit>Create SubVI



Note A red star on the Control Reference control indicates that the refnum is strictly typed. Refer to the *Strictly Typed and Weakly Typed Control Refnums* section of the *Controlling Front Panel Objects* topic of the *LabVIEW Help* for more information about weakly and strictly typed control references.

Creating Control References

To create a control reference for a front panel object, right-click the object or its block diagram terminal and select **Create>Reference** from the shortcut menu.

You can wire this control reference to a generic Property Node. You can pass the control reference to a subVI using a control refnum terminal.

Using Control References

Setting properties with a control reference is useful for setting the same property for multiple controls. Some properties apply to all classes of controls, such as the Disabled property. Some properties are only applicable to certain control classes, such as the Lock Boolean Text in Center property.

The following example shows how to construct a VI that uses a control reference on the subVI to set the Enable/Disable state of a control on the main VI front panel.

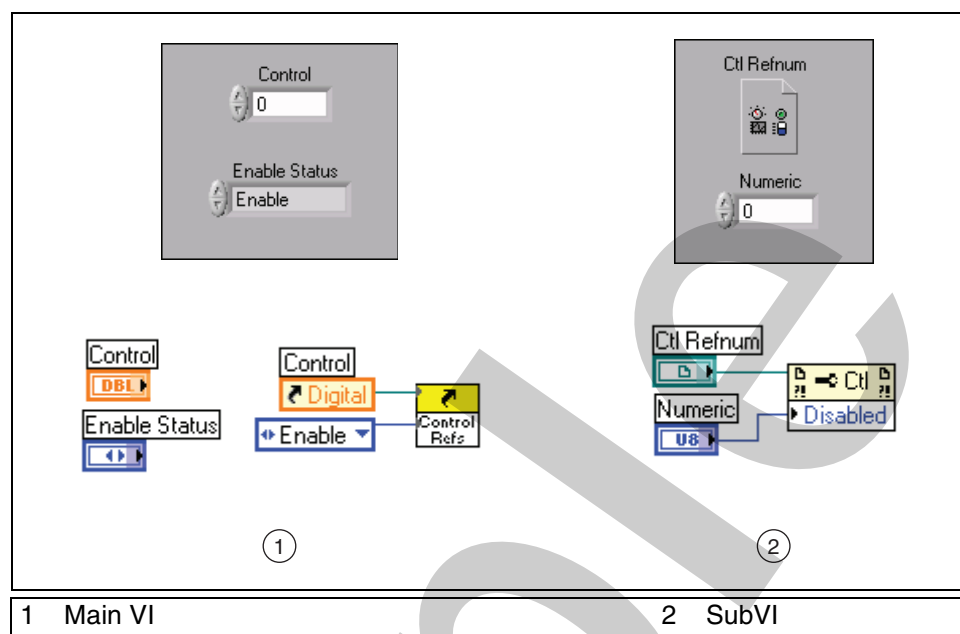


Figure 5-7. Control References

The main VI sends a reference for the digital numeric control to the subVI along with a value of zero, one, or two from the enumerated control. The subVI receives the reference by means of the **Ctl Refnum** on its front panel window. Then, the reference is passed to the Property Node. Because the Property Node now links to the numeric control in the main VI, the Property Node can change the properties of that control. In this case, the Property Node manipulates the enabled/disabled state.

Notice the appearance of the Property Node in the block diagram. You cannot select a property in a generic Property Node until the class is chosen. The class is chosen by wiring a reference to the Property Node. This is an example of an explicitly linked Property Node. It is not linked to a control until the VI is running and a reference is passed to the Property Node. The advantage of this type of Property Node is its generic nature. Because it has no explicit link to any one control, it may be reused for many different controls. This generic Property Node is available on the **Functions** palette.

Selecting the Control Type

When you add a Control Refnum to the front panel of a subVI, you next need to specify the VI Server Class of the control. This specifies the type of control references that the subVI will accept. In the previous example, Control was selected as the VI Server Class type, as shown in Figure 5-7. This allows the VI to accept a reference to any type of front panel control.

However, you can specify a more specific class for the refnum to make the subVI more restrictive. For example, you can select Digital as the class, and the subVI only can accept references to numeric controls of the class

Digital. Selecting a more generic class for a control refnum allows it to accept a wider range of objects, but limits the available properties to those that apply to all objects which the Property Node can accept.

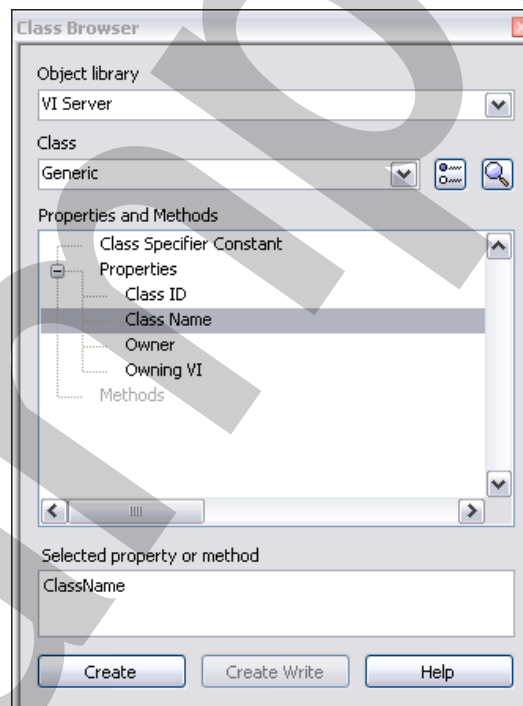
To select a specific control class, right-click the control and select **Select VI Server Class»Generic»GObject»Control** from the shortcut menu. Then, select the specific control class.

Creating Properties and Methods with the Class Browser Window

You can use the Class Browser window to select an object library and create a new property or method.

Complete the following steps to create a new property or method using the Class Browser window.

1. Select **View»Class Browser** to display the Class Browser window.



2. From the Object library pull-down menu, select a library.
3. Select a class from the Class pull-down menu. Use the following buttons to navigate the classes.



- Click the **Select View** button to toggle between an alphabetical view and a hierarchical view of the items in the Class pull-down menu and the Properties and Methods list.



- Click the **Search** button to launch the Class Browser Search dialog box.

4. From the Properties and Methods list in the Class Browser window, select a property or method. The property or method you select appears in the Selected property or method box.
5. Click the **Create** button or the **Create Write** button to attach a node with the selected property or method to your mouse cursor and add the node to the block diagram. The Create button creates a property for reading or a method. This button dims when you select a write-only property. To create a property for writing, click the **Create Write** button. The Create Write button dims when you select a method or read-only property. You also can drag a property or method from the Properties and Methods list directly to the block diagram.
6. Repeat steps 2 through 5 for any other properties and methods you want to create and add to the block diagram.

To practice the concepts in this section, complete Exercise 5-3.

Self-Review: Quiz

1. For each of the following items, determine whether they operate on a VI class or a Control class.

- Format and Precision
- Blinking
- Reinitialize to Default Value
- Show Tool Bar



2. You have a ChartGraph control refnum, shown at left, in a subVI. Which of the following control references could you wire to the control refnum terminal of the subVI? (multiple answers)

- a. Control reference of an XY graph
- b. Control reference of a numeric array
- c. Control reference of a waveform chart
- d. Control reference of a Boolean control

Sample

Self-Review: Quiz Answers

1. For each of the following items, determine whether they operate on a VI class or a Control class.

- Format and Precision: **Control**
- Blinking: **Control**
- Reinitialize to Default Value: **Control**
- Show Tool Bar: **VI**



2. You have a GraphChart control refnum, shown at left, in a subVI. Which control references could you wire to the control refnum terminal of the subVI?

- Control reference of an XY graph**
- Control reference of a numeric array
- Control reference of a waveform chart**
- Control reference of a Boolean control

Notes

Sample